

吕高旭 编著



Visual C#

范例精要解析



- 提供最完备的C#学习资料，适合没有任何编程基础的入门级读者
- 对200多个工程范例逐行分析、手把手辅导，由浅入深引领，直至学会为止
- 以微软MVP视角深入阐述面向对象编程的精髓，讲解生动，令人跃跃欲试
- 摆脱窗口束缚，利用Console工程领略全套底层类库功能，代码可随意移植
- 用崭新的方式重新梳理可视化编程思想，洞悉事件、代理、绘图机制
- 全面覆盖集合、泛型、线程、数据流、网络、序列化、数据库等高级话题
- 创作和编辑细致入微，配合光盘源代码，使新手、老手都能成为行家里手



清华大学出版社

1 前言

Visual C#范例精要解析

吕高旭 编 著

清华大学出版社

北 京

内 容 简 介

本书通过大量的范例,在详细讲解 C# 基本语法的基础上,介绍利用 C# 开发 .NET 应用程序所需了解的各种技巧,并完整地加入了 C# 2.0 的新增功能,由浅入深,带领读者逐步迈向专业 C# 程序设计师之路。

本书适合想要用 C# 开发 .NET 应用程序的读者阅读,也可供大专院校软件专业师生参考。

说明:经原作者同意,本书的第 24 章由清华大学出版社编辑宋延清先生增补创作。其余各章均系原作者的繁体版改写的简体中文版。

本书繁体字版书名为《Visual C# 2005 精要剖析》,由文魁资讯股份有限公司出版,版权属吕高旭所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号 图字:01-2007-5696

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

Visual C# 范例精要解析/吕高旭编著. —北京:清华大学出版社, 2008.1

ISBN 978-7-302-16586-6

I. V… II. 吕… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 188398 号

责任编辑:张彦青 宋延清

封面设计:杨玉兰

版式设计:北京东方人华科技有限公司

责任印制:孟凡玉

出版发行:清华大学出版社 地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:40 字 数:951 千字

附光盘 1 张

版 次:2008 年 1 月第 1 版 印 次:2008 年 1 月第 1 次印刷

印 数:1~4000

定 价:58.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026660-01

序 言

C#是一种用来开发 .NET 应用程序的语言，它以 C 语言为基础，延用其大部分的关键字、表达式以及运算符，但语法设计却更为简洁易懂，除避免了其他程序设计语言的缺点外，还新增了各种出色的语言特质，例如垃圾回收机制、类型安全以及异常处理等，最重要的是，它完全支持面向对象的程序设计方式。C#结合 .NET 框架所提供的类库，改变了传统应用程序的开发模式，其面向对象特性加上微软全力发展的 .NET 平台，使其在不远的将来，会成为一种最重要的程序设计语言。

1. 关于本书

本书除了讲解 C#的基本语法外，还将讲述利用 C#开发 .NET 应用程序所需了解的各种技巧，通过大量的范例，详细地说明各种相关技术的实际运用。此外还完整地加入了 C# 2.0 的新增功能，例如泛型、静态类及 Null 类型等。全书涵盖 C#语言基础、类和对象、.NET 类库以及图形接口程序设计等知识，由浅入深，带领读者逐步迈向专业 C#程序设计之路。

2. 内容结构

本书可以分成如下 4 个部分。

- C#语言基础(第 1~6 章): 介绍 C#编辑工具 Visual C# 2005 Express Edition 以及 .NET 框架和 C#基本语法元素。
- 类和对象(第 7~12 章): 介绍类、面向对象理论及运算符重载技术、程序错误捕获及异常处理机制。
- .NET 类库(第 13~20 章): 探讨 .NET 框架类库、核心命名空间、文字处理、集合与泛型应用、IO 和网络及序列化技术。
- 图形接口设计(第 21~24 章): 介绍代理及事件机制、窗口应用程序、GDI+及绘图技术、数据库应用程序。

3. 软件需求

本书使用 Visual C# 2005 Express Edition 撰写各章节的范例程序，读者也可以使用功能更加强大的 Microsoft Visual Studio 2005，编程界面和效果完全相同(附书光盘中的全部 C#项目均在这两种环境中测试通过)。

4. 适合对象

本书针对想要用 C#开发 .NET 应用程序的读者，内容涵盖了最基础的语法元素，即使没有编程基础的初学者在学习本书时也不会遇到任何困难，而已经具备 C++、Java 或 Visual Basic 等程序语言背景的读者在学习过程中会更加轻松。

目 录

第 1 章 C#概述	1	3.3.1 整数类型	29
1.1 .NET 框架与 C#	1	3.3.2 浮点类型	30
1.2 类库(Class Library)	2	3.3.3 decimal 类型	32
1.3 Visual C# 2005 Express	3	3.3.4 char 和转义字符	32
1.3.1 安装 Express	3	3.3.5 布尔类型	34
1.3.2 Express 的帮助文件	7	3.4 类型转换	34
1.3.3 创建和执行项目	8	3.5 枚举类型	36
1.4 C#应用程序类型	12	3.6 引用类型	38
1.5 本章习题	13	3.7 栈(Stack)和堆(Heap)	39
第 2 章 初探 C#	14	3.8 本章习题	40
2.1 第 1 个 C#应用程序	14	第 4 章 运算符	42
2.1.1 创建执行文件	14	4.1 表达式	42
2.1.2 程序结构	15	4.1.1 操作数和运算符	42
2.1.3 注释	16	4.1.2 运算符分类	43
2.1.4 类声明	16	4.2 赋值与算术运算	44
2.1.5 程序的入口点 Main()	17	4.2.1 赋值运算符	44
2.1.6 关键字	17	4.2.2 算术运算符	45
2.1.7 变量和数据类型	18	4.2.3 复合运算符和自增自减运算符	46
2.1.8 程序语句	18	4.2.4 类型转换错误	49
2.2 类和命名空间	19	4.3 关系运算和逻辑运算	50
2.2.1 关于类	19	4.3.1 关系运算符	50
2.2.2 命名空间	20	4.3.2 逻辑运算符与短路运算符	51
2.3 基本输入/输出	21	4.3.3 逻辑逐位运算符	53
2.4 本章习题	22	4.3.4 移位运算符	55
第 3 章 变量和数据类型	24	4.4 三元运算符	57
3.1 使用变量	24	4.5 运算符的优先级	58
3.1.1 变量声明	24	4.6 本章习题	59
3.1.2 变量的命名	25	第 5 章 流程控制及循环	61
3.1.3 赋值和初始化	25	5.1 流程控制	61
3.1.4 设置常量	27	5.1.1 if 语句	61
3.2 数据类型	28	5.1.2 if 嵌套	65
3.3 值类型	28	5.1.3 if-else-if 语句	66

5.1.4 switch 语句	68	第 8 章 扩展类	132
5.2 循环	74	8.1 对象的创建和销毁	132
5.2.1 for 语句	74	8.1.1 对象的创建和引用	132
5.2.2 while 语句	76	8.1.2 构造方法	133
5.2.3 do-while 语句	78	8.1.3 构造方法参数和多重构造 方法	135
5.2.4 跳转语句	80	8.1.4 销毁方法	137
5.3 变量的生存期	82	8.2 访问修饰符	137
5.4 本章习题	83	8.2.1 关于修饰符	137
第 6 章 数组	84	8.2.2 public 与 private	138
6.1 关于数组	84	8.3 属性成员	141
6.1.1 数组的类型	84	8.3.1 设计属性成员	141
6.1.2 定义数组	85	8.3.2 属性限制	144
6.1.3 数组索引	86	8.4 索引器	146
6.1.4 访问数组元素	86	8.4.1 使用索引器	146
6.1.5 数组的特性	88	8.4.2 多维索引器	149
6.1.6 使用 foreach	90	8.5 静态成员	153
6.2 二维数组	91	8.6 静态类	156
6.2.1 矩形数组	91	8.7 结构	157
6.2.2 锯齿形数组	94	8.8 嵌套类	159
6.3 矩阵相乘	96	8.9 本章习题	161
6.4 魔术矩阵	99	第 9 章 运算符重载	163
6.5 本章习题	103	9.1 运算符重载的概念	163
第 7 章 类和方法	105	9.2 实现运算符重载	164
7.1 类(Class)	105	9.3 二元运算符重载	166
7.1.1 关于类	105	9.4 处理不同类型之间的运算	169
7.1.2 定义类	106	9.5 逻辑运算符重载	172
7.1.3 类实例及成员引用	108	9.6 重载关系运算符	177
7.2 方法函数成员(Method)	110	9.7 转换运算符	184
7.2.1 方法	110	9.8 本章习题	188
7.2.2 定义方法	112	第 10 章 面向对象与继承	189
7.2.3 方法的返回值	115	10.1 面向对象的概念	189
7.2.4 参数传递	118	10.2 继承(Inheritance)	191
7.2.5 传址参数与 ref	121	10.2.1 关于继承	191
7.2.6 返回值与 out	122	10.2.2 实现继承	192
7.3 使用 this 关键字	123	10.2.3 继承结构的访问限制	198
7.4 方法重载	125	10.2.4 限制继承(Sealed Class)	199
7.5 递归(Recursion)	127		
7.6 本章习题	130		

10.3 覆写	200	第 13 章 C#核心命名空间	272
10.3.1 覆写方法成员	200	13.1 数学运算	272
10.3.2 使用 base 和 new 关键字	203	13.1.1 自然对数常数和圆周率 常数	272
10.3.3 使用 new 创建新方法	206	13.1.2 三角函数	273
10.4 构造方法的继承	208	13.1.3 自然对数与指数函数	274
10.5 Object 类	214	13.1.4 数值比较运算	275
10.5.1 Object 类的成员	214	13.1.5 近似值运算	276
10.5.2 覆写 Object 的函数	216	13.1.6 绝对值运算	277
10.5.3 Boxing 与 Unboxing	216	13.1.7 除法运算	277
10.6 本章习题	217	13.2 取得随机数	277
第 11 章 抽象类和接口	219	13.3 数值类型结构	280
11.1 抽象类(Abstract Class)	219	13.3.1 整数类型结构	281
11.2 接口	225	13.3.2 浮点数类型结构	283
11.2.1 定义与使用接口	225	13.3.3 Boolean	284
11.2.2 将派生类实例对象转型 为接口类型	227	13.3.4 Decimal	285
11.2.3 定义接口属性	233	13.3.5 Char	285
11.3 扩展接口	234	13.4 System.Array 类	287
11.3.1 接口继承	235	13.5 本章习题	289
11.3.2 实现多重接口	237	第 14 章 字符串处理	291
11.3.3 指定实现的接口	239	14.1 字符串(String)类	291
11.4 本章习题	241	14.1.1 认识字符串	291
第 12 章 异常处理	243	14.1.2 创建 String 类对象	292
12.1 异常处理概述	243	14.1.3 String 类的属性成员	293
12.1.1 程序异常	243	14.1.4 字符串查看和比较	294
12.1.2 异常处理	243	14.1.5 分割字符串与获取 子字符串	297
12.2 异常处理机制	245	14.1.6 字符删除、插入及大小写 转换	300
12.2.1 使用 try-catch	245	14.1.7 合并字符串	303
12.2.2 异常类(Exception)	248	14.2 动态字符串(StringBuilder)类	304
12.2.3 精确地捕捉异常	253	14.3 格式化字符串	307
12.2.4 使用 finally	257	14.3.1 格式化概述	307
12.2.5 嵌套 try-catch 区块	258	14.3.2 自定义数值格式	309
12.3 throw 语句	261	14.3.3 日期时间格式化	312
12.4 自定义异常类	265	14.3.4 ToString 方法	316
12.5 查看异常类	267	14.4 本章习题	317
12.6 本章习题	268		

第 15 章 集合	319	17.3.1 判断线程的结束	378
15.1 System.Collections 命名空间	319	17.3.2 取得线程状态	380
15.2 列举接口	320	17.4 避免资源访问冲突	381
15.3 ICollection 接口和实现类	324	17.4.1 lock 语句	381
15.3.1 ICollection 接口	325	17.4.2 Monitor 类	384
15.3.2 堆栈与 Stack 类	325	17.5 终止线程	388
15.3.3 队列与 Queue 类	328	17.6 线程管理(Thread Pool)	391
15.3.4 BitArray 类	331	17.7 死锁(Deadlock)	394
15.4 IList 接口	334	17.8 本章习题	394
15.4.1 IList 接口成员	335	第 18 章 IO 与数据流	396
15.4.2 ArrayList 类	335	18.1 IO 类概述	396
15.4.3 方法 Sort()与对象排序	338	18.1.1 数据流的概念	396
15.5 IDictionary 接口	340	18.1.2 IO 类	397
15.5.1 哈希与 HashTable 类	340	18.2 文件目录操作	398
15.5.2 二元搜索与 SortList 类	343	18.2.1 与文件目录有关的类	398
15.5.3 IDictionaryEnumerator 接口	345	18.2.2 操作目录	399
15.6 IComparable 接口	347	18.2.3 操作文件	402
15.7 本章习题	350	18.3 数据流	408
第 16 章 泛型	352	18.3.1 读写字节数据	408
16.1 C#与泛型	352	18.3.2 内存数据流 (MemoryStream)	411
16.1.1 集合的未定类型	352	18.3.3 文件数据流 FileStream	414
16.1.2 设计泛型类	354	18.3.4 提升数据读写性能——使用 缓冲数据流	416
16.2 列举	357	18.4 字符数据(Text Data)读写	418
16.2.1 实现 IEnumerable<T>	357	18.4.1 TextReader 类	419
16.2.2 使用 Iterator	361	18.4.2 TextWriter 类	419
16.3 default 关键字	363	18.4.3 使用派生类	420
16.4 Nullable 类型	364	18.5 随机访问	422
16.5 本章习题	366	18.6 异步 IO	424
第 17 章 线程	368	18.7 隔离存储	426
17.1 线程	368	18.8 本章习题	432
17.2 应用线程	368	第 19 章 网络技术	433
17.2.1 Thread 类	369	19.1 地址与 DNS	433
17.2.2 创建线程	370	19.2 Socket 应用程序	437
17.2.3 暂停线程——使用 Sleep() 和 Join()	374	19.2.1 命名空间 System.Net.Sockets	437
17.3 线程状态	378	19.2.2 实现 Socket 应用程序的	

要点	438
19.2.3 创建 TCP 联机	438
19.2.4 客户端应用程序	440
19.2.5 服务器端应用程序	443
19.2.6 网络数据流	447
19.3 Web 数据流	448
19.3.1 System.Net 命名空间	448
19.3.2 URI 与 Uri 类	449
19.3.3 WebRequest 和 WebResponse	450
19.3.4 对 HTTP 通信协议的支持	453
19.3.5 WebClient 类	457
19.4 本章习题	460
第 20 章 序列化(Serialization)	461
20.1 序列化类	461
20.2 自定义序列化行为	466
20.3 序列化属性的继承	469
20.4 修正无法序列化的数据	471
20.5 本章习题	474
第 21 章 事件与代理	476
21.1 关于事件	476
21.2 关于代理	478
21.2.1 代理类型	478
21.2.2 善用代理	480
21.3 事件处理	483
21.3.1 代理与事件机制	483
21.3.2 事件处理程序	484
21.3.3 EventArgs 类型自变量	486
21.3.4 内置的代理类型——事件 处理器(EventHandler)	489
21.4 多重传送代理(Multicasting)	492
21.5 事件的多重处理	493
21.6 匿名方法	497
21.7 本章习题	499
第 22 章 Windows 应用程序	501
22.1 创建窗体	501
22.1.1 创建 Form 对象	501
22.1.2 使用 Express 创建 Windows 应用程序	502
22.1.3 窗体的属性	506
22.2 事件处理	508
22.2.1 窗体事件	509
22.2.2 事件处理器 EventHandler	511
22.2.3 键盘事件	512
22.2.4 鼠标事件	516
22.2.5 Paint 事件	520
22.3 消息对话框(Message Box)	522
22.4 控件	525
22.5 在可视化环境中使用控件	527
22.6 标签和文本框	528
22.7 按钮与 ButtonBase 类	533
22.7.1 按钮(Button)	534
22.7.2 CheckBox、RadioButton 及 GroupBox	538
22.8 ListBox 和 ComboBox	539
22.8.1 ListBox 控件	539
22.8.2 ComboBox 控件	545
22.9 本章习题	549
第 23 章 绘图技术	551
23.1 关于 GDI+	551
23.2 绘图基础及 Graphics 对象	551
23.2.1 使用 Graphics 类	551
23.2.2 坐标系统	554
23.2.3 Point 结构数据类型	555
23.2.4 Pen 类	555
23.2.5 Color 结构数据类型	556
23.3 图形绘制	556
23.3.1 绘制矩形和多边形	556
23.3.2 绘制弧线和椭圆	559
23.4 文字输出	564
23.4.1 绘制字符串	564
23.4.2 Font 与 FontFamily	566
23.4.3 StringFormat 类型对象	568
23.5 绘制曲线	572
23.5.1 连接曲线点	572
23.5.2 绘制一般曲线	573

23.5.3 贝兹曲线(Bezier Spline)	578	24.2 访问数据库的步骤.....	604
23.6 路径和裁剪区域.....	582	24.3 访问 Access 数据库演示.....	606
23.6.1 路径(Path).....	582	24.3.1 设置 Access 数据库.....	606
23.6.2 转换路径	585	24.3.2 建立操作数据库的简单 程序.....	611
23.7 笔刷(Brush).....	588	24.4 深入理解 ADO.NET	615
23.7.1 SolidBrush 类.....	589	24.4.1 什么是 ADO.NET	615
23.7.2 HatchBrush 类	590	24.4.2 向 DataSet 中填充数据	617
23.7.3 渐变	594	24.4.3 归纳 DataAdapter 的用法.....	618
23.7.4 运用 PathGradientBrush	599	24.4.4 使用 DataGridView 控件	619
23.8 本章习题	601	24.5 本章习题.....	624
第 24 章 编写数据库应用程序.....	603	附录	625
24.1 OleDb 简介	603		

第 1 章 C#概述

C#是一种面向对象的编程语言，专门用来开发执行在 .NET 平台上的应用程序，整个 C#语言体系都构建在 .NET 框架之上。本章将从 .NET 框架基础知识开始，说明 C#与 .NET 框架之间的关系。

1.1 .NET 框架与 C#

在正式介绍 C#之前，我们首先简要说明 .NET 框架和 C#的基础知识，你很难只是单独学习 C#而不去了解什么是 .NET 框架。二者紧密结合，才使 C#程序设计人员能够通过 .NET 框架轻易地开发出功能强大的应用程序。

.NET 框架是微软公司开发的一种全新的运行平台，它提供 Windows 操作系统执行应用程序所需的服务及相关的应用程序接口，图 1.1 简要地说明了其组成架构。

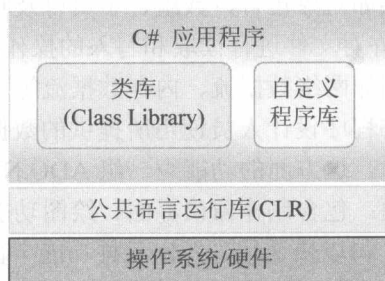


图 1.1 .NET 框架的组成架构

.NET 框架包含了两个主要部分，公共语言运行库(Common Language Runtime)以及类库(Class Library)，分别说明如下。

- 公共语言运行库(Common Language Runtime): 简称为 CLR，提供执行 .NET 应用程序所需的管理机制，例如多线程管理、内存管理以及类型检查等。应用程序的源代码在这样的机制下，确保了其安全性和正确性。

CLR 是 .NET 框架的基础，托管程序代码的运作，在 CLR 上所执行的程序代码被称为受托管(Managed)的程序代码。利用 C#编写的程序一般必须在 CLR 的环境下执行，受 CLR 监控，所以 C#应用程序本身均为受托管的程序代码。

其他不是在 CLR 的环境下执行的程序代码，称为非托管(Unmanaged)的程序代码，例如 .NET 出现之前的 VB6、C++等应用程序均属于这一类程序代码。

- 类库(Class Library): 包含数量庞大的类，这些类是以面向对象理论为基础而设计的，提供创建各种应用程序所需的广泛功能，从基本的 IO 数据流、丰富的图形界面 API、多线程应用程序的支持，到复杂的分布式应用程序设计，程序开发人员几乎都可以直接从类库中找到相关的解决方案。

在学习 C#语言的过程中, 你会发现 .NET 框架已经构建好创建应用程序所需的基础, 当 C#语言的基本语法学习告一段落之后, 就可以把大部分时间放在了解如何使用类库所提供的各种功能方面。

1.2 类库(Class Library)

类库包含了功能广泛的各种类(Class), 类是 .NET 应用程序的基础元素; 类库中所包含的类根据其功能大致可以分为以下几种。

- 提供基础数据处理功能: 其中包含了类之间的转换操作、格式化以及字符串数据处理等相关功能。
- 提供应用程序的异常处理机制: 在应用程序执行期间进行一致性的错误处理, 包括异常返回以及错误的捕获。
- 在执行期查看加载的类信息: 提供执行期间查看应用程序加载类信息和创建的类实例(Instance)的功能, 并且针对这些执行期间动态产生的类实例进行调用(Invoke)和存取操作。
- 封装数据结构: 提供如对象集合、数组、字典以及哈希表等数据结构的实现。
- 创建 IO 数据流: 允许对文件进行读取和写入的操作。针对各种存储体, 支持字节数据的读写操作, 例如网络数据流、内存数据流等。
- 进行数据访问: 允许程序设计人员使用所提供的数据库访问服务, 进行各种类型数据库的数据维护操作, 这方面的功能由一组 ADO.NET 组件来完成。
- 创建图形化用户界面: 包含制作窗口界面、绘图功能的相关类。

C#借助类库的强大支持, 得以快速地开发出各种功能丰富的应用程序, 当你继续往下阅读本书的各个章节时, 将会发现除了与 C#基本语法以及面向对象理论相关的探讨外, 其他章节所涉及的内容都与类库所提供的特定类密切关联。

类库提供编写应用程序所需功能的各种类, 创建应用程序的过程, 通常是创建特定类的对象, 然后引用其方法和属性成员的过程, 如图 1.2 所示。

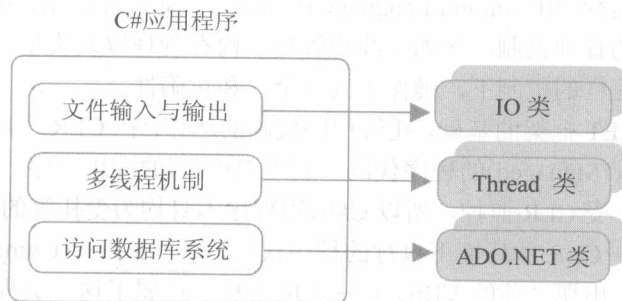


图 1.2 创建应用程序的过程

例如当你想要开发一个具有多线程功能的应用程序时, 只需直接引用类库所提供的 Thread 类即可完成所需的功能; 又如创建图形窗口界面时, 必须借助 Form 类以及相关的

可视化控件来完成；甚至当你想要开发一套数据库系统时，类库也包含了一组功能强大的 ADO.NET 类库供开发人员创建数据库应用程序所需的完整功能。

C#使用 CLR 提供的服务，创建解决特定问题的应用程序；由于其完全面向对象的特性以及与类库的紧密结合，使得开发功能强大的商业级应用程序更为容易。

1.3 Visual C# 2005 Express

在进行 C#应用程序开发之前，首先必须安装 .NET 框架，它提供完成相关操作所需的基础环境，例如 C#编译器、类库以及程序执行环境等，可以直接从微软的官方网站下载完整组件。

从 C# 2.0 开始，我们可以不用自行下载 .NET 框架组件了，微软提供了开发 C#应用程序所需的专用可视化工具——Visual C# 2005 Express(速成版)，它是一种轻量级、包含编写程序代码所需各种功能的图形界面编辑器，.NET 框架也直接整合在其中，本书可以利用它来学习 C#，此后将把该工具简称为“Express”。

1.3.1 安装 Express

Express 可以直接从微软的官方网站下载，下载解压后，单击一个名称为 setup.exe 的安装文件图标，如图 1.3 所示，即可开始安装程序。

首先出现如图 1.4 所示的【欢迎使用安装程序】界面，直接单击【下一步】按钮，进入安装程序。



图 1.3 setup.exe 图标

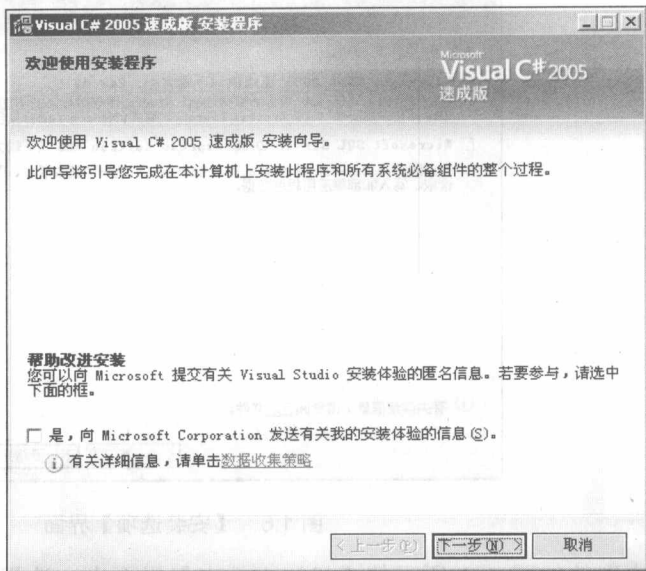


图 1.4 【欢迎使用安装程序】界面

在出现的如图 1.5 所示的【最终用户许可协议】界面中，勾选左下方的【我接受许可协议中的条款】，然后单击【下一步】按钮。

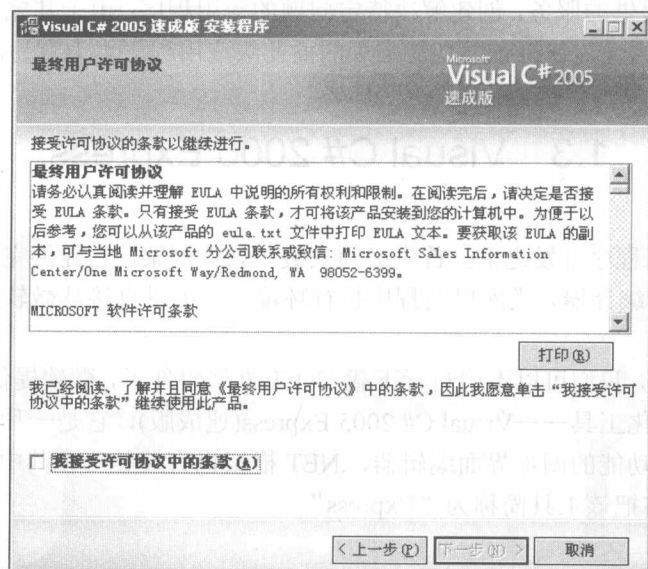


图 1.5 【最终用户许可协议】界面

在出现的如图 1.6 所示的【安装选项】界面中，读者可以选择是否安装 MSDN。MSDN 是关于 .NET 和 C# 的重要的官方说明文件。将其勾选，单击【下一步】按钮继续。

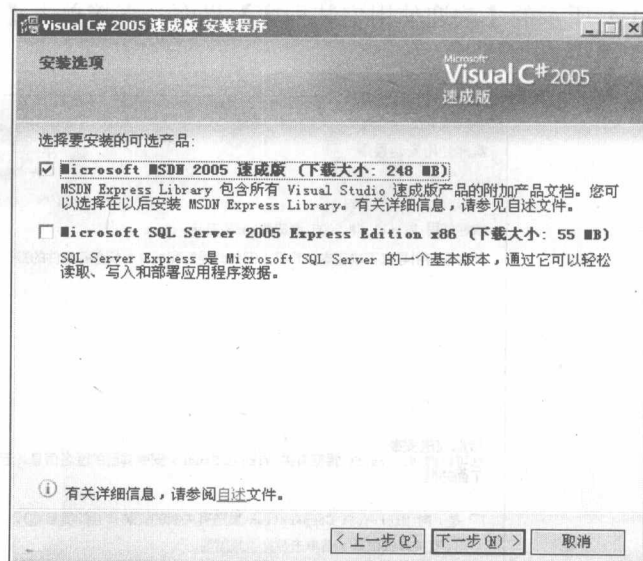


图 1.6 【安装选项】界面

在出现的如图 1.7 所示的【目标文件夹】界面中，选择安装路径，这里接受默认选项即可，单击【安装】按钮，即可开始自动安装程序。

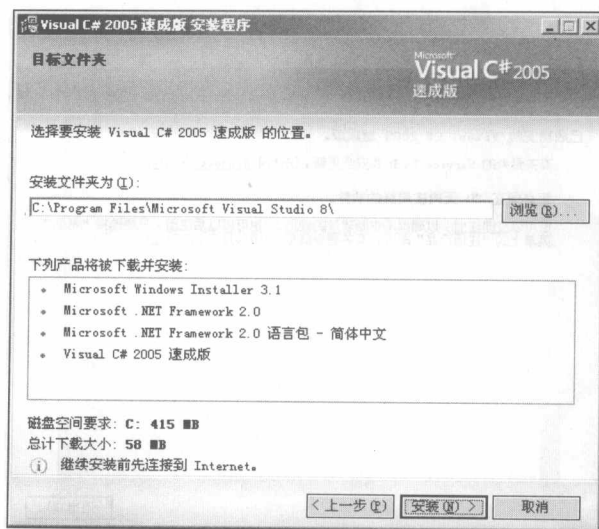


图 1.7 【目标文件夹】界面

当出现如图 1.8 所示的【下载和安装进度】界面时，表示安装程序已经开始进行，这个部分可能要花一点时间，等待其完成所有的安装即可。

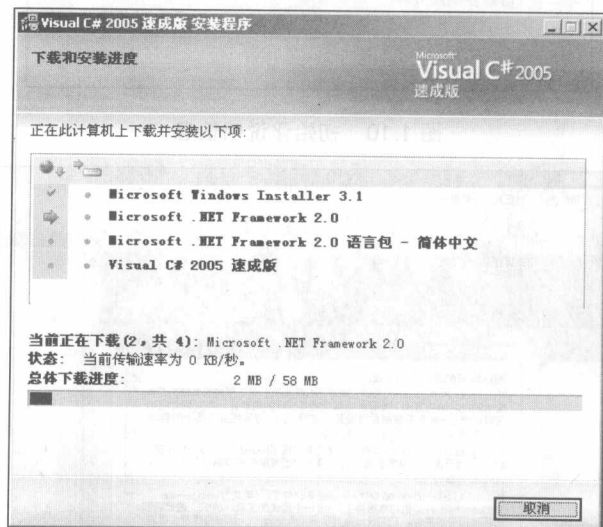


图 1.8 【下载和安装进度】界面

当所有的项目安装完成后，会出现如图 1.9 所示的【安装完成】界面，单击【退出】按钮即可。

Express 安装完成之后，就可以开始编写 C#应用程序。从 PC 桌面选择【开始】|【所有程序】|【Microsoft Visual C# 2005 Express 版】命令启动 Express，当第一次使用 Express 的时候，会出现如图 1.10 所示的初始化说明信息，稍候等它完成。

系统完成初始化操作之后，将出现 Express 的起始页，如图 1.11 所示。看到这个画面，表明已经完成了编写 C#应用程序所需的准备工作。

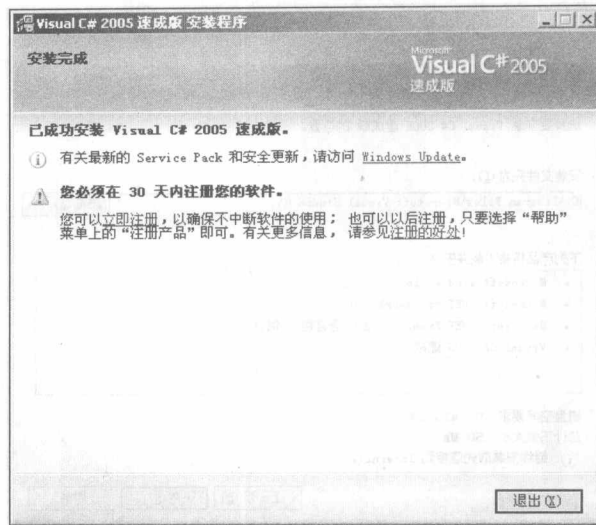


图 1.9 【安装完成】界面

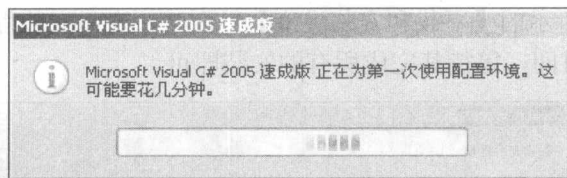


图 1.10 初始化说明信息

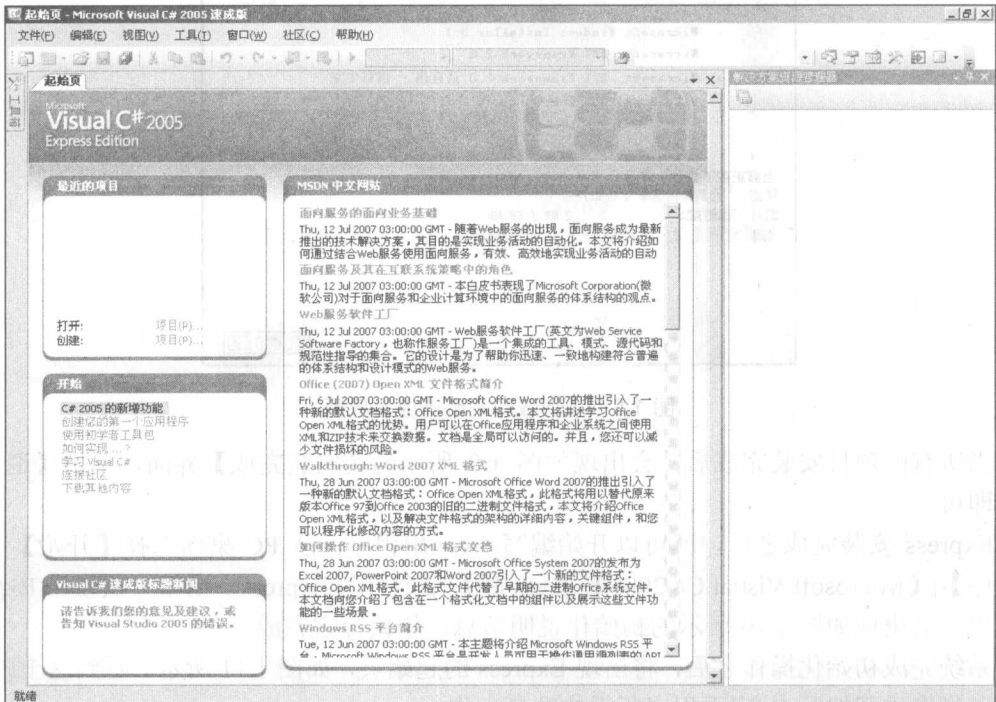


图 1.11 Express 的起始页

1.3.2 Express 的帮助文件

启动 Express 之后,单击【帮助】菜单,展开其下拉菜单,选择其中的【目录】命令,如图 1.12 所示。

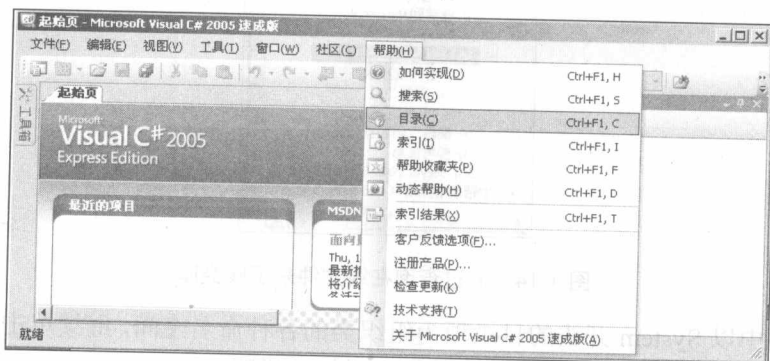


图 1.12 选择【目录】命令

将出现 MSDN 在线帮助文件,如图 1.13 所示,左边是帮助文件的目录,右边则是文件的内容。

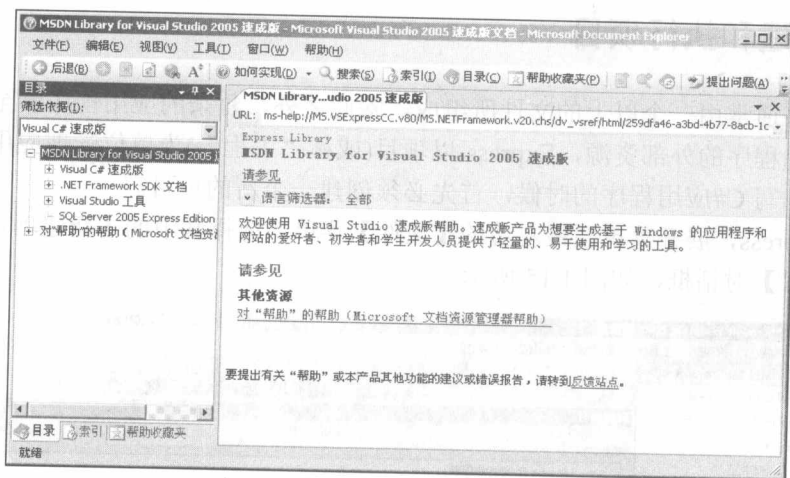


图 1.13 MSDN 在线帮助文件

展开【MSDN Library for Visual Studio 2005 速成版】节点,其中出现的项目为各种帮助文件的标题,可以继续展开各个子节点,查看其内容。

在开始学习 C#时,了解如何使用这份在线帮助文件非常重要,其中详细地记载了开发各种应用程序时所需了解的相关细节,而最重要的当然是关于类库的介绍——其中的类是组成类库的核心。

类库非常庞大,没有任何一本书可以完整地涵盖其中的全部内容,若需特定类的使用信息,可以直接查询此在线文件(见图 1.14),展开其中的类库目录,便可以看到所有的类组织列表。